83

AF/2800

WINSTEAD

OIPE JAN 08 2004 JC95 PATENT & TRADEMARK OFFICE

Docket No.: 40532-P001M10P2

January 5, 2004

Mail Stop AF- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Appellants:	Peter D. Geiger; Manuel J. Alvarez II; Thomas A. Dye		
Title:	Parallel Compression And Decompression System And Method Having Multiple Parallel Compression And Decompression Engines		
Docket No.:	40532-P001M10P2		
Serial No.:	10/044,786	Filing Date:	1/11/02
Examiner:	Howard L. Williams	Group Art Unit:	2819

RECEIVED JAN 13 2004 TC 2800 MAIL ROOM

Transmitted herewith are the following:

1. Return Receipt Postcard
2. This Transmittal Letter (2 pages) (in duplicate)
3. 46 pages of Appeal Brief Under 37 C.F.R. § 1.191 **(in triplicate)**

for the above-identified Application.

☒ Appeal Brief fee $165.00

☒ If the fee has been calculated incorrectly, the Commissioner is hereby authorized to charge any insufficiency of payment of the following fees associated with this communication, or credit any overpayment, to Deposit Account No. 23-2426 (40532-P001M10P2) in the name of Winstead Sechrest & Minick PC. **A duplicate copy of this transmittal letter is enclosed.**

CERTIFICATION UNDER 37 C.F.R. § 1.8
I hereby certify that this correspondence (along with any item referred to as being enclosed herewith) is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF– Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on January 5, 2004.

Toni Stanley

Toni Stanley

Respectfully submitted,

Michael Adams

Michael P. Adams
Attorney for Applicant(s)
Reg. No. 34,763

MPA:tls
Enclosure(s)

AUSTIN_1\239669\1
40532-P001M10P2 01/05/2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Peter D. Geiger; Manuel J. Alvarez II; Thomas A. Dye		
Assignee:	Quickshift, Inc. (f/k/a Interactive Silicon, Inc.)		
Title:	Parallel Compression And Decompression System And Method Having Multiple Parallel Compression And Decompression Engines		
Serial No.:	10/044,786	Filing Date:	1/11/02
Examiner:	Howard L. Williams	Group Art Unit:	2819
Docket No.:	40532-P001M10P2 (f/k/a 5143-03000)	Conf. No.:	9379

Dallas, Texas
January 5, 2004

Mail Stop AF - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 1.191

Dear Sir:

Appellants submit this Appeal Brief pursuant to the Notice of Appeal filed in this case on October 30, 2003 and received in the Patent Office on or about November 3, 2003. Thus, this Appeal Brief is due January 3, 2004, but extended to January 5, 2004 since January 3 was a Saturday. Enclosed is a check in the amount of $165.00, being the amount specified in 37 C.F.R. 1.17(c) for this Appeal Brief. The Commissioner is also authorized to deduct any other amounts required for this Appeal Brief and to credit any amounts overpaid to Deposit Account No. 23-2426. **This Brief is submitted in triplicate.**

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Quickshift, Inc., as named in the caption above. Please note that on December 29, 2003 Appellants filed for recording documentation which reflects that Quickshift, Inc. is the current assignee.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-119 are pending in the application. Claims 1-8, 10-25, 27-38, 40-43, 45-48, 50-61, and 63-119 stand rejected. Claims 9, 26, 39, 44, 49, and 62 were objected to.

IV. STATUS OF AMENDMENTS

The Appellants' response and claim amendments to the Office Action dated December 26, 2002, having a mailing date of March 26, 2003, have been considered, but the Examiner indicated that they did not place the application in condition for allowance because the Appellants' arguments were deemed unpersuasive. The Appellants' response to the Final Office Action dated April 30, 2003, having a mailing date of June 30, 2003, has also been considered, but the Examiner again indicated that it did not place the application in condition for allowance because the Appellants' arguments were deemed unpersuasive.

V. SUMMARY OF THE INVENTION

Multiple separate, serial compression and decompression engines running in parallel are cost prohibitive for general use servers, workstations, desktops, or mobile units.

Lower cost semiconductor devices have been developed that use compression hardware. However, these devices do not operate fast enough to run at memory speed and thus lack the necessary performance for in-memory data. Such compression hardware devices are limited to serial operation at compression rates that work for slow I/O devices such as tape backup units.

It is desirable to provide a method of increasing the effective size of system memory without increasing actual physical memory, and to thus allow processors and/or I/O masters of the system to address more system memory than physically exists. It is also desirable to improve data transfer bandwidth in computer systems.

Embodiments of a compression/decompression (codec) system may include a plurality of data compression engines each implementing a different data compression algorithm. The codec system may be designed for the reduction of data bandwidth and storage requirements and for compressing / decompressing data.

In one embodiment, uncompressed data may be compressed using a plurality of compression engines operating concurrently, *i.e.*, in parallel, with each engine compressing the data using a different lossless data compression algorithm. At least one of the data compression engines may implement a parallel lossless data compression algorithm. In one embodiment, each of the data compression engines implements a different parallel lossless data compression engine. Each parallel compression engine may have its own history buffer.

In one embodiment, each of the data compression engines implements a parallel lossless data compression algorithm. For example, each of the data compression engines may implement a parallel version of an LZ-based data compression algorithm. In this example, a first compression engine may implement a parallel LZ-based data compression algorithm which uses tag-based encoding, while a second compression engine may implement a parallel LZ-based data compression algorithm which uses "escape" characters to differentiate among compressed and "raw" data characters (and which does not use a tag-based coding scheme). In another embodiment, a first compression engine may implement a parallel LZ-based (or dictionary based) data compression algorithm, a second compression engine may implement a parallel RLE (run length encoding)

algorithm, and a third compression engine may implement a parallel Huffman-based encoding scheme.

The parallel compression engine of the present invention may implement an improved system and method for performing parallel data compression designed to process stream data at more than a single byte or symbol (character) at one time. The parallel compression engine may examine a plurality of symbols in parallel, thus providing greatly increased compression and decompression performance. More specifically, a parallel compression engine may operate to examine each of a plurality of symbols with each of a plurality of entries in a history buffer concurrently, *i.e.*, in a single clock cycle. The codec system may also include a parallel decompression engine that operates to examine a plurality of compressed symbols concurrently, *i.e.*, in a single clock cycle and may operate to produce a plurality of decompressed symbols concurrently, *i.e.*, per clock cycle. These parallel compression and decompression engines may be referred to herein as parallel codec engines. In one embodiment, the parallel compression and decompression engines may implement a modified single stream dictionary based (or history table based) data compression and decompression method, such as an LZ based method, to provide a scalable, high bandwidth compression and decompression operation.

The integrated data compression and decompression capabilities of the codec system removes system bottlenecks and increases performance. This allows lower cost systems due to smaller data storage requirements and reduced bandwidth requirements. This also increases system bandwidth and hence increases system performance. Thus the present invention provides a significant advance over the operation of current devices, such as memory controllers, memory modules, processors, and network devices, among others.

VI. ISSUES

A. Are claims 1-119 properly rejected under 35 U.S.C. § 112, first paragraph?

B. Are claims 1-11, 13-24, and 26-27 properly rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,311,159 to Van Tichelen, et al. (hereinafter "Van Tichelen")?

C. Are claims 1-4, 10-14, 17-21, 28-34, 45-46, 50-58, 61, 65-68, 76-78, 81, 85 and 100-102 properly rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,729,228 to Franaszek et al. ("Franaszek") or U.S. Patent No. 5,109,226 to MacLean, Jr. et al. ("MacLean")?

D. Are claims 1-5, 10-15, 17-22, 28-34, 45-46, 50-59, 61, 65-69, 76-79, 81, 85, and 100-102 properly rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,304,197 B1 to Freking et al. ("Freking")?

E. Are claims 6-8, 16, 23-25, 27, 47-48, 60, 63-64, 70-75, 80, 82-84 and 103-112 properly rejected under 35 U.S.C. § 102(b) as being anticipated by Franaszek?

F. Are claims 5, 15, 22, 59, 69, and 79 properly rejected under 35 U.S.C. § 102(b) as being anticipated by MacLean?

G. Are claims 35-38, 40-43, and 86-99 properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Franaszek in view of U.S. Patent No. 5,608,396 to Cheng et al. ("Cheng")?

H. Are claims 113-119 properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Franaszek?

VII. GROUPING OF THE CLAIMS

Claims 1-4, 10-14, 17-21, 28-34, 45-46, 50-58, 61, 65-68, 76-78, 81, 85 and 100-102 form a first group.

Claims 5, 15, 22, 59, 69 and 79 form a second group.

Claims 6-8, 16, 23-25, 27, 47-48, 60, 63-64, 70-75, 80, 82-84 and 103-112 form a third group.

Claims 35-38, 40-43, and 86-99 form a fourth group.

Claims 113-119 form a fifth group.

Claims 9, 26, 39, 44, 49 and 62 form a sixth group.

The reasons for these groupings are set forth in Appellants' arguments in Section VIII.

VIII. ARGUMENT

A. Introduction

Claims 1-119 were rejected under 35 U.S.C. § 112, first paragraph, as containing subject matter which allegedly was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventors at the time the invention was filed, had possession of the claimed invention. In particular, the Examiner alleges that no support in the specification as originally filed could be found for the claim amendments adding the language "operates independently." For the reasons discussed below, Appellants submit that there is support for these amendments, and, accordingly, Appellants respectfully traverse this rejection.

Claims 1-4, 10-14, 17-21, 28-34, 45-46, 50-58, 61, 65-68, 76-78, 81, 85 and 100-102 were rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,729, 228 to Franaszek et al.

("Franaszek") or U.S. Patent No. 5,109,226 to MacLean, Jr. et al. ("MacLean"). Claims 1-5, 10-15, 17-22, 28-34, 45-46, 50-59, 61, 65-69, 76-79, 81, 85, and 100-102 were rejected under 35 U.S.C. 102(e) as anticipated by U.S. Patent No. 6,304,197 B1 to Freking et al. ("Freking"). Claims 6-8, 16, 23-25, 27, 47-48, 60, 63-64, 70-75, 80, 82-84 and 103-112 were rejected under 35 U.S.C. 102(b) as anticipated by Franaszek. Claims 5, 15, 22, 59, 69, and 79 were rejected under 35 U.S.C. § 102(b) as anticipated by MacLean. Claims 35-38, 40-43, and 86-99 were rejected under 35 U.S.C. § 103(a) as unpatentable over Franaszek in view of U.S. Patent No. 5,608,396 to Cheng et al. ("Cheng"). Claims 113-119 were rejected under 35 U.S.C. § 103(a) as unpatentable over Franaszek. For the reasons discussed below, Appellants respectfully traverse these rejections.

Claims 9, 26, 39, 44, 49, and 62 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

B. Prior Art and Section 112 Rejections

Each of the rejected claims was rejected either under § 102 in view of Franaszek, Freking, or MacLean, or § 103 in view of Franaszek individually or in combination with Cheng. For the reasons discussed below, Appellants respectfully traverse these claim rejections and submit that Franaszek, Freking, and MacLean, either individually or in combination with the other cited patents, do not teach Appellants' claimed invention.

Appellants submit that their claimed invention can be distinguished from Franaszek and MacLean for at least the reason that each of the parallel compression engines in Appellants' claimed invention <u>operate independently</u> of the other engines. On the other hand, the operation of each of the compression engines in Franaszek and MacLean is <u>dependent</u> on the other engines, and thus they do not operate independently.

In Franaszek, the compression engines, referred to as "compressors," "*cooperatively construct* a dynamic compression dictionary and compress the sub-blocks in parallel using the dictionary." *See* Abstract and Fig. 2 (emphasis added). *See also* Col. 1, lines 48-50 of Franaszek (emphasis added): "The present invention alleviates the above-described problem with dictionary type coding by introducing *cooperation among the compressors*." Furthermore, Franaszek states: "Since the compressed sub-blocks were encoded using a *logically shared dictionary*, a logically shared dictionary 345 is required for the decompression." Col. 3, lines 26-28 (emphasis added). Thus, as discussed in more detail below, unlike Appellants' claimed invention, the compression engines in Franaszek use a "logically shared dictionary" and therefore do <u>not</u> operate independently.

Through the following statement in the Examiner's September 16, 2003 Advisory Action, the Examiner apparently admits that the compressors in Franaszek do not "operate independently" as in Appellant's claimed invention: "Franaszek may indeed build a shared dictionary; however, it seems that he does so to avoid the inefficiencies of dividing the stream and having it operated upon by truly <u>independent</u> operation of a plurality of same type algorithm compressors." (Emphasis added.) Thus, the Examiner acknowledges that the compressors of Franaszek do not each have their own dictionary which would enable independent operation.

Similarly, each of the compression engines of MacLean, referred to as "compaction processors," is dependent on the other engines/processors. In MacLean, the data to be compacted is divided into equal sized sets of data for each compaction processor. *See* Col. 2, lines 58-60. Each compaction processor must process the data directed to it in a known maximum amount of time. *See* Abstract. Also: "Each additional compaction processor affects the statistics format of the entire compaction process." Col. 5, lines 34-35. *See also* Fig. 1. Furthermore, the timing of the processing of each compaction processor is dependent on the timing of the preceding compaction processor in a sequence. For example, MacLean states that: "the signal A-mout transfers the control from one CP

[compaction processor] unit to the next. On the write cycle, which is when the data stream from the host is compacted and written onto the tape media, the stream of data is directed to all storage devices in the A section of the compaction processor. The storage device 136 signals the event counter 132, that one set of data has been received, *i.e.*, 512 bytes. The event counter 132, in turn, signals the interface control 134 unit that CP1 [compaction processor 1] has received its section of data, *i.e.*, one set, and the next set of data should be received by CP2 [compaction processor 2]." *See* Col. 7, line 24 to Col. 8, line 18. For at least these reasons, MacLean does not teach Appellants' claimed invention.

The Examiner asserts that the above referenced teachings of MacLean do not show less independence between the compression engines of MacLean versus the compression engines of Appellants' claimed invention. Appellants respectfully disagree. In particular, Appellants note that the Examiner has not adequately addressed the teachings of MacLean that state that "Each additional compaction processor affects the statistics format of the entire compaction process." Col. 5, lines 34-35. *See also* Fig. 1. As is discussed in more detail below, Appellants' compression engines operate independently for at least the reason that they do not use a logically shared dictionary, but have their own independent dictionaries which do not affect the other dictionaries. Particularly in view of the above quoted language from MacLean that notes that "Each additional compaction processor affects the statistics format of the entire compaction process," the Examiner has not shown any teachings in MacLean reflecting that the compaction processors have their own independent dictionaries. To the contrary, MacLean appears to teach away from the concept of each compaction processor having its own independent dictionary, especially in view of the above-quoted passage and the fact that the data is divided into equal sized sets of data for each compaction processor, each compaction processor must process the data directed to it in a known maximum amount of time and the timing of each processor is dependent upon the timing of the preceding processor.

Appellants also submit that the Freking patent is not relevant to Appellants' claimed invention. Freking does not pertain to a system or method for compressing respective portions of data in parallel. Rather, Freking pertains to data compression wherein after compression the resulting data elements may be separated and processed in parallel. Freking discusses the variable-length coding (VLC) digital signal processing technique which is often used to compress data. Freking notes that in the prior art there is no discernible demarcation between the data elements resulting from the VLC technique. However, Freking describes a means of separating and processing multiple data elements in parallel after VLC compression. *See* Col. 1, lines 18-61, Col. 4, lines 64-65, and Col. 5, lines 15-45. Freking does not describe compressing such data in parallel.

The Examiner asserts that Freking is relevant to Appellants' claimed invention in that it discloses variable length encoders operating in parallel. The Examiner cites to Col. 8, lines 20-63 of Freking for support. Appellants note that the passage the Examiner cited from Freking does not teach "parallel compression engines, wherein each of the plurality of engines operates independently and implements a parallel data compression algorithm." Accordingly, the Examiner has not provided adequate support of his claim rejections in view of the Freking patent.

As discussed above, the compression engines in Appellants' claimed invention operate independently for at least the reason that they do not require use of a logically shared dictionary and instead have their own independent dictionary. For example, Appellants' patent application states that: "In one embodiment, each parallel compression engine 570A, 570B, 570C and 570D has its own history buffer." Page 13, lines 4-5. Appellants' application further states that:

> In one embodiment, each of the parallel compression engines may implement a different type of parallel dictionary based (or LZ-based) compression. For example, a first parallel compression engine may implement a parallel LZ compression scheme according to or similar to U.S. Patent No. 6,208,273 using tag-based encoding techniques, wherein tag bits are used to differentiate among compressed and "raw" (uncompressed) data characters or symbols. In this example, a second

> parallel compression engine may implement a parallel LZ compression scheme according to or similar to U.S. Patent No. 6,208,273 using an escape character (or sequence) to indicate the beginning of (or differentiate between) compressed and "raw" data streams (and not using tag-based encoding techniques), etc. A third parallel compression engine may implement a parallel LZ compression scheme according to or similar to U.S. Patent No. 6,208,273, wherein the history buffer is pre-loaded with all 256 possible symbols, and thus all received data is compressed (with a pointer to a previous entry in the history buffer). In this third parallel compression engine, tags or escape characters are not required to differentiate among compressed and raw data characters, as all characters or symbols are compressed (with a pointer to a previous entry in the history buffer).

Page 18, lines 5-20.

Appellants respectfully submit that the Examiner has not provided any support in the cited prior art for teachings which reflect compression engines operating independently, especially in the manner set forth in the above excerpts from Appellants' application.

For at least these reasons, Appellants submit that their independent claims are allowable over Franaszek, Freking, and MacLean, either individually or in combination with the other cited patents. Appellants further submit that the dependent claims are allowable for at least these same reasons.

Furthermore, Appellants note that at least the above-cited passages reflect that the specification conveys with reasonable clarity to those skilled in the art that Appellants were indeed in possession of the claimed invention at the time the application was filed. *See* MPEP § 2163.02. Accordingly, Appellants respectfully traverse the Examiner's § 112, paragraph one rejection.

Appellants respectfully submit that the Examiner has not satisfied the examiner's burden of proof for establishing rejection of claims 1-119 under 35 U.S.C. § 112, paragraph one for lack of written description. As stated in MPEP § 2163:

> A description as filed is presumed to be adequate, unless or until sufficient evidence or reasoning to the contrary has been presented by the examiner to rebut the presumption. See, e.g., *In re Marzocchi*, 439

> F.2d 220, 224, 169 USPQ 367, 370 (CCPA 1971). The examiner, therefore, must have a reasonable basis to challenge the adequacy of the written description. The examiner has the initial burden of presenting by a preponderance of evidence why a person skilled in the art would not recognize in an applicant's disclosure a description of the invention defined by the claims. *Wertheim*, 451 F.2d at 263, 191 USPQ at 97. In rejecting a claim, the examiner must set forth express findings of fact regarding the above analysis which support the lack of written description conclusion. These findings should:
>
> (A) Identify the claim limitation at issue; and
>
> (B) Establish a *prima facie* case by providing reasons why a person skilled in the art at the time the application was filed would not have recognized that the inventor was in possession of the invention as claimed in view of the disclosure of the application as filed. A general allegation of "unpredictability in the art" is not a sufficient reason to support a rejection for lack of adequate written description.

See also MPEP § 2163.02:

> Whenever the issue arises, the fundamental factual inquiry is whether the specification conveys with reasonable clarity to those skilled in the art that, as of the filing date sought, applicant was in possession of the invention as now claimed. See, e.g., *Vas-Cath, Inc. v. Mahurkar*, 935 F.2d 1555, 1563-64, 19 USPQ2d 1111, 1117 (Fed. Cir. 1991). ... The subject matter of the claim need not be described literally (i.e., using the same terms or *in haec verba*) in order for the disclosure to satisfy the description requirement.

With regard to the rejection of claims 1-119, the Examiner did not provide any express findings of fact, reasons or any other support for the 35 U.S.C. § 112, paragraph one rejection. The Examiner merely stated that no support in the specification as originally filed could be found for the claim amendment which added the phrase "operates independently."

Despite the Examiner not providing any reasons or otherwise satisfying the burden for a 35 U.S.C. § 112, paragraph one rejection, in their June 30, 2003 response to the final office action, Appellants did cite to disclosure in the specification which supports the "operates independently" claim amendment. However, in the advisory action, the Examiner merely repeated the 35 U.S.C. §

112, paragraph one rejection, and again provided no reasons or support for this rejection. According to MPEP § 2163:

> Upon reply by applicant, before repeating any rejection under 35 U.S.C. § 112, para. 1, for lack of written description, review the basis for the rejection in view of the record as a whole, including amendments, arguments, and any evidence submitted by applicant. If the whole record now demonstrates that the written description requirement is satisfied, do not repeat the rejection in the next office action. If the record still does not demonstrate that the written description is adequate to support the claim(s), repeat the rejection under 35 U.S.C. 112, para. 1, fully respond to applicant's rebuttal arguments, and properly treat any further showings submitted by applicant in the reply.

Thus, in the advisory action the Examiner again did not supply the required findings or support for the 35 U.S.C. 112, paragraph one rejection.

Since the Examiner has not satisfied the burden of proof for a 35 U.S.C. § 112, paragraph one rejection, and the Appellants provided unrebutted support for their claim amendments, the rejection of claims 1-119 should not stand.

IX. CONCLUSION

For the above reasons, Appellant respectfully submits that rejection of pending Claims 1-119 is unfounded. Accordingly, Appellant requests that the rejection of Claims 1-119 be reversed.

This Brief is submitted in triplicate.

Respectfully submitted,

Michael Adams

Michael P. Adams
Attorney for Appellant(s)
Reg. No. 34,763

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence (along with any item referred to as being enclosed herewith) is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on January 5, 2004.

Toni Stanley

Toni Stanley

APPENDIX

1. A data compression system comprising:

a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a parallel data compression algorithm;

wherein each of the plurality of parallel compression engines is operable to:

receive a different respective portion of uncompressed data; and

compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and

output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data.

2. The data compression system of claim 1,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

3. The data compression system of claim 1,

wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

4. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.

5. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.

6. The data compression system of claim 1, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

7. The data compression system of claim 6, wherein each of the plurality of parallel compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.

8. The data compression system of claim 6, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

9. The data compression system of claim 6,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

10. The data compression system of claim 1, wherein the parallel data compression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW

algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential compression algorithm.

11. The data compression system of claim 1, further comprising:

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm;

wherein each of the plurality of parallel compression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data.

12. The data compression system of claim 11,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

13. The data compression system of claim 11,

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

14. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.

15. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.

16. The data compression system of claim 11, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

17. The data compression system of claim 11, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols,

wherein each of the plurality of parallel decompression engines is operable to:

receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

18. (Amended) A data compression system comprising:

a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a parallel data compression algorithm;

first logic coupled to the plurality of parallel compression engines and configured to:

receive uncompressed first data; and

provide a different respective portion of the uncompressed first data to each of the plurality of parallel compression engines;

wherein each of the plurality of parallel compression engines is configured to:

compress the different respective portion of the uncompressed first data using the parallel data compression algorithm to produce a compressed portion of the first uncompressed data; and

output the compressed portion of the first uncompressed data;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of compressed portions of the first uncompressed data.

19. The data compression system of claim 18,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed first data to produce the compressed portions of the first uncompressed data.

20. The data compression system of claim 18, further comprising:

second logic coupled to the plurality of parallel compression engines and configured to:

receive the plurality of compressed portions of the data; and

merge the plurality of compressed portions of the first data to produce compressed first data.

21. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.

22. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.

23. The data compression system of claim 18, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

24. The data compression system of claim 23, wherein each of the plurality of parallel compression engines implements a parallel data compression algorithm based on a Lempel-Ziv (LZ) algorithm.

25. The data compression system of claim 23, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

26. The data compression system of claim 23,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed first data, wherein the uncompressed first data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

27. The data compression system of claim 18, wherein the parallel data compression algorithm is based on a serial dictionary-based data compression algorithm.

28. The data compression system of claim 18, wherein the parallel data compression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential compression algorithm.

29. The data compression system of claim 18, further comprising:

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm;

third logic coupled to the plurality of parallel decompression engines and configured to:

receive compressed second data; and

provide a different respective portion of the compressed second data to each of the plurality of parallel decompression engines;

wherein each of the plurality of parallel decompression engines is configured to:

decompress the different respective portion of the compressed second data to produce an uncompressed portion of the second data; and

output the uncompressed portion of the compressed second data;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of uncompressed portions of the compressed second data.

30. The data compression system of claim 29,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to produce the plurality of uncompressed portions of the compressed second data.

31. The data compression system of claim 29, further comprising:

fourth logic coupled to the plurality of parallel decompression engines and configured to:

receive the plurality of uncompressed portions of the compressed second data; and

merge the plurality of uncompressed portions of the compressed second data to produce uncompressed second data.

32. (Amended) A data compression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;

first logic coupled to the plurality of compression engines and configured to:

receive uncompressed data;

provide a different portion of the uncompressed data to each of the plurality of compression engines;

wherein each of the plurality of compression engines is configured to compress a received uncompressed portion of the data to produce a compressed portion of the data, wherein, in said compressing, each of the plurality of compression engines is configured to:

maintain a history table comprising entries, wherein each entry comprises at least one symbol;

receive the uncompressed portion of the data, wherein the uncompressed portion of the data comprises a plurality of symbols;

compare the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing produces compare results;

determine match information for each of the plurality of symbols based on the compare results; and

output the compressed portion of the data in response to the match information.

33. The data compression system of claim 32, wherein said compressing is performed by the plurality of compression engines in a parallel fashion to produce a plurality of compressed portions of the data.

34. The data compression system of claim 33, further comprising

second logic coupled to the plurality of compression engines and configured to:

receive the plurality of compressed portions of the data from the plurality of compression engines; and

merge the plurality of compressed portions of the data to produce compressed data.

35. (Amended) A memory controller, comprising:

memory control logic for controlling a memory; and

a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a lossless parallel data compression algorithm;

wherein each of the plurality of parallel compression engines is operable to:

receive a different respective portion of uncompressed data; and

compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and

output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data;

wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

36. The memory controller of claim 35,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

37. The memory controller of claim 35, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

38. The memory controller of claim 35, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

39. The memory controller of claim 35,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

40. (Amended) A memory module, comprising:

one or more memory devices for storing data; and

a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a lossless parallel data compression algorithm;

wherein each of the plurality of parallel compression engines is operable to:

receive a different respective portion of uncompressed data; and

compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and

output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data;

wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

41. The memory module of claim 40,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

42. The memory module of claim 40, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

43. The memory module of claim 40, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

44. The memory module of claim 40,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

45. (Amended) A network device, comprising:

network logic for performing networking functions; and

a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a lossless parallel data compression algorithm;

wherein each of the plurality of parallel compression engines is operable to:

receive a different respective portion of uncompressed data; and

compress the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data; and

output the respective compressed portion;

wherein the plurality of parallel compression engines are configured to perform said compression in a parallel fashion to produce a plurality of respective compressed portions of the uncompressed data;

wherein the respective compressed portions output from the plurality of parallel compression engines are combinable to form compressed data corresponding to the uncompressed data.

46. The network device of claim 45,

wherein, in performing said compression in a parallel fashion, the plurality of parallel compression engines operate concurrently to compress the different respective portions of the uncompressed data to produce the compressed portions of the uncompressed data.

47. The network device of claim 45, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

48. The network device of claim 45, wherein the uncompressed data comprises a plurality of symbols, wherein each of the plurality of parallel compression engines is operable to compare each of a plurality of received symbols with each of a plurality of entries in a history table concurrently.

49. The network device of claim 45,

wherein each of the plurality of parallel compression engines comprises:

an input for receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

a history table comprising entries, wherein each entry comprises at least one symbol;

a plurality of comparators for comparing the plurality of symbols with entries in the history table, wherein the plurality of comparators are operable to compare each of the plurality of symbols with each entry in the history table concurrently, wherein the plurality of comparators produce compare results;

match information logic coupled to the plurality of comparators for determining match information for each of the plurality of symbols based on the compare results, wherein the match information logic is operable to determine if a contiguous

match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

an output coupled to the match information logic for outputting compressed data in response to the match information.

50. (Amended) A data compression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;

first logic coupled to the plurality of compression engines and configured to:

receive uncompressed data; and

provide a different portion of the uncompressed data to each of the plurality of compression engines;

wherein each of the plurality of compression engines is configured to:

compress the uncompressed portion of the uncompressed data provided to the particular compression engine to produce a compressed portion of the uncompressed data; and

output the compressed portion of the uncompressed data;

wherein the plurality of compression engines are configured to perform said compressing in a parallel fashion to produce a plurality of compressed portions of the uncompressed data in parallel; and

second logic coupled to the plurality of compression engines and configured to:

receive the plurality of compressed portions of the uncompressed data; and

combine the plurality of compressed portions of the uncompressed data to produce compressed data.

51. The system of claim 50, wherein the system further comprises:

a processor;

a memory coupled to the processor and to the second logic and configured to store data for use by the processor;

wherein the second logic is further configured to write the compressed data to the memory.

52. (Amended) A system comprising:

a processor;

a memory coupled to the processor and operable to store data for use by the processor;

a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm; and

first logic coupled to the memory and to the plurality of compression engines and configured to:

receive uncompressed first data;

split the uncompressed first data into a plurality of uncompressed portions of the first data; and

provide the plurality of uncompressed portions of the uncompressed first data to the plurality of compression engines; and

wherein the plurality of compression engines are configured to operate concurrently to compress the plurality of uncompressed portions of the uncompressed first data to produce a plurality of compressed portions of the uncompressed first data.

53. The system of claim 52, further comprising

second logic coupled to the plurality of compression engines and to the memory and configured to merge the plurality of compressed portions of the uncompressed first data to produce compressed first data;

wherein the second logic is further configured to write the compressed first data to the memory.

54. The system of claim 52, further comprising:

a plurality of decompression engines;

third logic coupled to the memory and to the plurality of decompression engines and configured to:

receive compressed second data;

split the compressed second data into a plurality of compressed portions of the compressed second data;

provide the plurality of compressed portions of the compressed second data to the plurality of decompression engines; and

wherein the plurality of decompression engines are configured to operate concurrently to decompress the plurality of compressed portions of the compressed second

data to produce a plurality of uncompressed portions of the compressed second data.

55. The system of claim 54, wherein each of the plurality of decompression engines implements a parallel data decompression algorithm

56. The system of claim 54, further comprising:

fourth logic coupled to the plurality of decompression engines and configured to combine the plurality of uncompressed portions of the compressed second data to produce uncompressed second data.

57. (Amended) A method for compressing data, the method comprising:

receiving uncompressed data;

providing a different respective portion of the uncompressed data to each of a plurality of parallel compression engines, wherein each of the plurality of parallel compression engines operates independently and implements a parallel data compression algorithm;

each of the plurality of parallel compression engines compressing the different respective portion of the uncompressed data using the parallel data compression algorithm to produce a respective compressed portion of the uncompressed data, wherein the plurality of parallel compression engines operate concurrently to perform said compressing in a parallel fashion, wherein the plurality of parallel compression engines produce a plurality of respective compressed portions of the uncompressed data;

combining the plurality of respective compressed portions of the uncompressed data to produce compressed data, wherein the compressed data corresponds to the uncompressed data; and

outputting the compressed data.

58. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel lossless data compression algorithm.

59. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel statistical data compression algorithm.

60. The method of claim 57, wherein each of the plurality of parallel compression engines implements a parallel dictionary-based data compression algorithm.

61. The method of claim 57, further comprising writing the compressed data to a memory.

62. The method of claim 57,

wherein, for each of the plurality of parallel compression engines, said compressing comprises:

receiving the different respective portion of the uncompressed data, wherein the uncompressed data comprises a plurality of symbols, wherein the plurality of symbols includes a first symbol, a last symbol, and one or more middle symbols;

maintaining a history table comprising entries, wherein each entry comprises at least one symbol;

comparing the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing in a parallel fashion comprises comparing each of the plurality of symbols with each entry in the history table concurrently, wherein said comparing produces compare results;

determining match information for each of the plurality of symbols based on the compare results, wherein said determining match information includes determining if a contiguous match occurs for one or more of the one or more middle symbols that does not involve a match with either the first symbol or the last symbol; and

outputting compressed data in response to the match information.

63. (Amended) A method comprising:

receiving uncompressed data;

providing a different portion of the uncompressed data to each of a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;

each of the plurality of compression engines compressing its respective different portion of the uncompressed data to produce a compressed portion of the data, wherein said compressing comprises:

maintaining a history table comprising entries, wherein each entry comprises at least one symbol;

receiving the respective different portion of the uncompressed data, wherein the respective different portion of the uncompressed data comprises a plurality of symbols;

comparing the plurality of symbols with entries in the history table in a parallel fashion, wherein said comparing produces compare results;

determining match information for each of the plurality of symbols based on the compare results; and

outputting the compressed portion of the data in response to the match information.

wherein said compressing is performed by the plurality of compression engines in a parallel fashion to produce a plurality of compressed portions of the uncompressed data.

64. The method of claim 63, further comprising:

merging the plurality of compressed portions of the uncompressed data to produce compressed data; and writing the compressed data to a memory.

65. (Amended) A data decompression system comprising:

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data.

66. The data decompression system of claim 65,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective

portions of the compressed data to produce the uncompressed portions of the compressed data.

67. The data decompression system of claim 65,

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

68. The data decompression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.

69. The data decompression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.

70. The data compression system of claim 65, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

71. The data compression system of claim 70, wherein each of the plurality of parallel decompression engines implements a parallel data decompression algorithm based on a Lempel-Ziv (LZ) algorithm.

72. The data decompression system of claim 70, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:

receive the different respective portion of the compressed data, wherein the different respective portion of the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the different respective portion of the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

73. The data decompression system of claim 72,

wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

74. The data decompression system of 73,

wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;

wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

75. The data decompression system of claim 72, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;

wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

76. (Amended) A data decompression system comprising:

a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

first logic coupled to the plurality of decompression engines and configured to:

receive compressed data; and

provide a different respective portion of the compressed data to each of the plurality of decompression engines;

wherein each of the plurality of decompression engines is configured to:

decompress the respective compressed portion of the compressed data to produce an uncompressed portion of the compressed data; and

output the uncompressed portion of the compressed data;

wherein the plurality of decompression engines are configured to operate concurrently to perform said decompressing in a parallel fashion to produce a plurality of uncompressed portions of the compressed data.

77. The data decompression system of claim 76, further comprising:

second logic coupled to the plurality of decompression engines and configured to:

receive the plurality of uncompressed portions of the compressed data; and

merge the plurality of uncompressed portions of the compressed data to produce uncompressed data.

78. The data decompression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel lossless data decompression algorithm.

79. The data decompression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel statistical data decompression algorithm.

80. The data compression system of claim 76, wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

81. The data decompression system of claim 76, wherein the parallel data decompression algorithm is based on one of an LZSS algorithm, an LZ77 algorithm, an LZ78 algorithm, an LZW algorithm, an LZRW1 algorithm, a Run Length Encoding (RLE) algorithm, a Predictive Encoding algorithm, a Huffman coding algorithm, an Arithmetic coding algorithm and a Differential decompression algorithm.

82. (Amended) A data decompression system comprising:

a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

first logic coupled to the plurality of decompression engines and configured to:

receive compressed data;

provide a different portion of the compressed data to each of the plurality of decompression engines;

wherein each of the plurality of decompression engines is configured to decompress its received different portion of the compressed data to produce an uncompressed portion

of the data, wherein, in said decompressing, each of the plurality of decompression engines is configured to:

receive the different portion of the compressed data, wherein the different portion of the compressed data comprises tokens each describing one or more uncompressed symbols;

examine a plurality of tokens from the different portion of the compressed data in parallel in a current decompression cycle;

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window; and

generate an uncompressed portion of the compressed data comprising the plurality of symbols using the plurality of selects.

83. The data decompression system of claim 82, wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the compressed data.

84. The data decompression system of claim 83, further comprising

second logic coupled to the plurality of decompression engines and configured to:

receive the plurality of uncompressed portions of the compressed data from the plurality of decompression engines; and

merge the plurality of uncompressed portions of the compressed data to produce uncompressed data.

85. (Amended) A data decompression system comprising:

a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

first logic coupled to the plurality of decompression engines and configured to:

receive compressed data; and

provide a different portion of the compressed data to each of the plurality of decompression engines;

wherein each of the plurality of decompression engines is configured to:

decompress the compressed portion of the data provided to the particular decompression engine to produce an uncompressed portion of the data; and

output the uncompressed portion of the data;

wherein the plurality of decompression engines is configured to perform said decompressing in a parallel fashion to produce a plurality of uncompressed portions of the data in parallel; and

second logic coupled to the plurality of decompression engines and configured to:

receive the plurality of uncompressed portions of the data; and

merge the plurality of uncompressed portions of the data to produce uncompressed data.

86. (Amended) A memory controller, comprising:

memory control logic for controlling a memory; and

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

87. The memory controller of claim 86,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective

portions of the compressed data to produce the uncompressed portions of the compressed data.

88. The memory controller of claim 86,

wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

89. The memory controller of claim 88, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:

receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

90. The memory controller of claim 89,

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

91. The memory controller of 90,

wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

92. The memory controller of claim 89, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;

wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

93. (Amended) A memory module, comprising:

at least one memory device for storing data; and

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

94. The memory module of claim 93,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

95. The memory module of claim 93,

wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

96. The memory module of claim 95, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:

receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

97. The memory controller of claim 96,

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

98. The memory controller of 97,

wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

99. The memory controller of claim 96, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;

wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

100. (Amended) A network device, comprising:

network logic for interfacing to a network; and

a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;

wherein each of the plurality of parallel decompression engines is operable to:

receive a different respective portion of compressed data; and

decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and

output the respective uncompressed portion;

wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data;

wherein the respective uncompressed portions output from the plurality of parallel decompression engines are combinable to form uncompressed data corresponding to the compressed data.

101. The network device of claim 100,

wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate concurrently to decompress the different respective portions of the compressed data to produce the uncompressed portions of the compressed data.

102. The network device of claim 100,

wherein each of the plurality of parallel decompression engines implements a parallel dictionary-based data decompression algorithm.

103. The network device of claim 102, wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein, in decompressing the different respective portion of the compressed data, each of the plurality of parallel decompression engines is operable to:

receive the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examine a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generate the uncompressed data comprising the plurality of symbols in response to said examining.

104. The network device of claim 103,

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

105. The network device of 104,

wherein each of the plurality of parallel decompression engines operates in a pipelined fashion;

wherein, in examining the plurality of tokens from the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.

106. The network device of claim 103, wherein each of the plurality of parallel decompression engines is further operable to:

generate a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window;

wherein each of the plurality of parallel decompression engines generates the uncompressed data using the plurality of selects.

107. (Amended) A method for decompressing data, comprising:

receiving compressed data;

providing a different portion of the compressed data to each of a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

each of the plurality of decompression engines decompressing the different portion of the compressed data, wherein said decompressing produces an uncompressed portion of the data, wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the compressed data; and

combining the plurality of uncompressed portions of the compressed data to produce uncompressed data.

108. The method of claim 107, further comprising writing the uncompressed data to a memory.

109. The method of claim 107, wherein the parallel data decompression algorithm is a parallel dictionary-based decompression algorithm.

110. The method of claim 107,

wherein the compressed data comprises a compressed representation of uncompressed data, wherein the uncompressed data has a plurality of symbols;

wherein each of the plurality of decompression engines decompressing the different portion of the compressed data comprises;

receiving the different portion of the compressed data, wherein the compressed data comprises tokens each describing one or more of the symbols in the uncompressed data;

examining a plurality of tokens from the compressed data in parallel in a current decompression cycle; and

generating the uncompressed data comprising the plurality of symbols in response to said examining.

111. (Amended) A method comprising:

receiving compressed data;

providing a different portion of the compressed data to each of a plurality of decompression engines, wherein each of the plurality of decompression engines operates independently and implements a parallel data decompression algorithm;

each of the plurality of decompression engines decompressing a compressed portion of the data provided to the particular decompression engine to produce an uncompressed portion of the data, wherein said decompressing comprises:

receiving the compressed portion of the data, wherein the compressed portion of the data comprises tokens each describing one or more uncompressed symbols;

examining a plurality of tokens from the compressed portion of the data in parallel in a current decompression cycle;

generating a plurality of selects in parallel in response to examining the plurality of tokens in parallel, wherein each of the plurality of selects points to a symbol in a combined history window; and

generating an uncompressed portion of the data comprising the plurality of symbols using the plurality of selects;

wherein said decompressing is performed by the plurality of decompression engines in a parallel fashion to produce a plurality of uncompressed portions of the data.

112. The method of claim 111, further comprising:

merging the plurality of uncompressed portions of the data to produce uncompressed data; and

writing the uncompressed data to a memory.

113. (Amended) A data compression/decompression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;

a plurality of decompression engines, wherein each of the plurality of decompression engines implements a parallel data decompression algorithm;

first logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:

receive data;

if the data is uncompressed, provide a plurality of uncompressed portions of the data to each of the plurality of data compression engines; and

if the data is compressed, provide a plurality of compressed portions of the data to each of the plurality of data decompression engines;

wherein, if the data is uncompressed, the plurality of compression engines are configured to compress the plurality of uncompressed portions of the data in a parallel fashion to produce a plurality of compressed portions of the data; and

wherein, if the data is compressed, the plurality of decompression engines are configured to decompress the plurality of compressed portions of the data in a parallel fashion to produce a plurality of uncompressed portions of the data.

114. The data compression/decompression system of claim 113, further comprising:

second logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:

if the data is uncompressed, merge the compressed portions of the data produced by the plurality of compression engines to produced compressed data; and

if the data is compressed, merge the uncompressed portions of the data produced by the plurality of decompression engines to produced uncompressed data.

115. The data compression/decompression system of claim 113, wherein the parallel data decompression algorithm and the parallel data decompression algorithm are based on a serial lossless data compression/decompression algorithm.

116. (Amended) A data compression/decompression system comprising:

a plurality of compression/decompression engines, wherein each of the plurality of compression/decompression engines operates independently and implements a parallel data compression algorithm and a parallel data decompression algorithm;

first logic coupled to the plurality of data compression/decompression engines and configured to:

receive data;

split the data into a plurality of portions of the data; and

provide the plurality of portions of the data to the plurality of data compression/decompression engines;

wherein the plurality of data compression/decompression engines is configured to:

if the data is uncompressed, compress the portions of the data in a parallel fashion to produce a plurality of compressed portions of the first data; and

if the data is compressed, decompress the portions of the data in a parallel fashion to produce a plurality of uncompressed portions of the first data.

117. The data compression/decompression system of claim 116, further comprising:

second logic coupled to the plurality of data compression/decompression engines and to the plurality of data decompression engines and configured to

if the data is uncompressed, merge the compressed portions of the data produced by the plurality of compression/decompression engines to produced compressed data; and

if the data is compressed, merge the uncompressed portions of the data produced by the plurality of compression/decompression engines to produced uncompressed data.

118. The data compression/decompression system of claim 116, wherein the parallel data decompression algorithm and the parallel data decompression algorithm are lossless parallel dictionary-based compression/decompression algorithms.

119. (Amended) A system comprising:

a processor;

a memory coupled to the processor and operable to store data for use by the processor;

a data compression/decompression system comprising:

a plurality of compression engines, wherein each of the plurality of compression engines operates independently and implements a parallel data compression algorithm;

a plurality of decompression engines, wherein each of the plurality of decompression engines implements a parallel data decompression algorithm;

first logic coupled to the plurality of data compression engines and to the plurality of data decompression engines and configured to:

receive first data;

if the first data is uncompressed, provide a plurality of uncompressed portions of the first data to each of the plurality of compression engines; and

if the first data is compressed, provide a plurality of compressed portions of the first data to each of the plurality of decompression engines;

wherein, if the first data is uncompressed, the plurality of compression engines is configured to compress the plurality of uncompressed portions of the first data in a parallel fashion to produce a plurality of compressed portions of the first data; and

wherein, if the first data is compressed, the plurality of decompression engines is configured to decompress the plurality of compressed portions of the first data in a parallel fashion to produce a plurality of uncompressed portions of the first data.

AUSTIN_1\239643\1
40532-P001M10P2 01/05/2004